A PROGRAM FOR MONTE CARLO STUDY OF RIDGE ESTIMATES IN CANONICAL ANALYSIS

by

E. J. Carney and D. A. Anderson

BU-520-M June, 1974

Introduction

Instability of estimates of canonical variates [5] and the similarity of canonical analysis to the regression problem suggest the possibility that ridge regression techniques [2,3] might improve stability of estimates of canonical variates. A program is described which may be used to generate many pseudo-random samples from a multivariate normal distribution with a given (input) correlation structure, and systematically compute the canonical variates and canonical correlations for a series of increments to the diagonals of the diagonal blocks of the sample correlation matrices. The program computes the averages and mean square errors of the estimated canonical correlations and canonical variates. Results of some experiments using the program, with a single correlation matrix are reported in [1]. Related work has been reported in [6].

Canonical analysis attempts to find, for two sets of random vectors, the linear combinations of the vectors in each set which have maximum correlation, then the linear combinations in each set, orthogonal to the first, which again maximize the correlation, etc. Let the true correlation matrix of a p-variate normal distribution be Σ. Let there be p_1 vectors in the first set and $p_2 \geq p_1$ vectors in the second set. The correlation matrix may be partitioned as

$$\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix},$$

where Σ_{11}, $p_1 \times p_1$, is the correlation matrix for the first set of variates, Σ_{22}, $p_2 \times p_2$, the correlation matrix of the second set, and $\Sigma_{12} = \Sigma'_{21}$, the correlations of the vectors in the first set, with the vectors in the second set. The canonical variates and canonical correlations may be obtained by computing the eigen vectors and eigen values of

$$T = \Sigma_{11}^{-\frac{1}{2}} \Sigma_{12} \Sigma_{22}^{-1} \Sigma_{21} \Sigma_{11}^{-\frac{1}{2}}$$

If B_1^* is a matrix whose rows are the eigen vectors of T, then

$$B_1^* T B_1^{*\prime} = D^2$$

where D is a diagonal matrix whose elements are the canonical correlations. The canonical variates for the first set are the rows of

$$B_1 = B_1^* \Sigma_{11}^{-\frac{1}{2}} .$$

If B_2^* is a matrix whose first p_1 rows are determined by

$$B_1^* \Sigma_{11}^{-\frac{1}{2}} \Sigma_{12} \Sigma_{22}^{-\frac{1}{2}} = [D^2 | 0] B_2^*$$

and whose remaining rows are any orthogonal completion, then the canonical variates of the second set are given by the rows of

$$B_2 = B_2^* \Sigma_{22}^{-\frac{1}{2}} .$$

Estimates of the canonical variates and canonical correlations may be obtained by applying the above operations to $\hat{\Sigma}$, the sample correlation matrix. These estimates will be denoted by $\hat{B}_1^*$, $\hat{B}_1$, etc. The ridge estimates are obtained by incrementing the diagonals of $\hat{\Sigma}_{11}$ and $\hat{\Sigma}_{22}$ by k_1 and k_2, non-negative real numbers, and obtaining eigen values and eigen vectors of

$$\hat{T}(k_1,k_2) = (\hat{\Sigma}_{11} + k_1 I)^{-\frac{1}{2}} \hat{\Sigma}_{12} (\hat{\Sigma}_{22} + k_2 I)^{-1} \hat{\Sigma}_{21} (\hat{\Sigma}_{11} + k_1 I)^{-\frac{1}{2}} .$$

The purpose of the Monte Carlo study for which the program was designed is to examine the effect upon the estimates $\hat{B}_1(k_1,k_2)$, $\hat{B}_2(k_1,k_2)$ and $\hat{D}(k_1,k_2)$ for

various values of k_1 and k_2, by comparing the estimated values with the true values obtained from a known correlation matrix.

Computer Programs

The Monte Carlo computations are performed by a FORTRAN program using several IMSL subroutines [4]. A brief outline of the operations performed is as follows:

1. Read program parameters:

   n - number of observations in a sample

   p_1 - number of variates in first set

   p_2 - number of variates in second set

   m - number of n x (p_1+p_2) multivariate samples to be generated

   seed - starting value for random number generator

   Several other parameters to control output options are also read.

2. Read the population correlation matrix, Σ.

3. Compute the population canonical analysis obtaining B_1^*, B_1, B_2^*, B_2 and D.

4. Read k_1 and k_2 values to be used as increments to the diagonals of $\hat{\Sigma}_{11}$ and $\hat{\Sigma}_{22}$.

5. Obtain a multivariate sample of n (p_1+p_2)-variate vectors from $N(0,\Sigma)$ and compute the sample correlation matrix.

6. Partition the correlation matrix and factor the diagonal blocks into triangular matrices by Cholesky decomposition.

7. Form the updated triangular matrices, obtain $\hat{T}(k_1,k_2)$ and perform the eigen value iteration to obtain the sample canonical variates and canonical correlations for each k_1,k_2 pair.

8. Accumulate the statistics for each k_1, k_2 pair: Averages out mean square errors of the canonical correlations and canonical variates.

9. Repeat steps 5-8 for m samples.

10. Print the resulting averages and mean square errors for each pair of k_1, k_2 values.

The listings of the FORTRAN routines used (except those which are IMSL subprograms) are appended. The following list gives the names of the subroutines used and a brief statement of their function. The list is arranged to indicate the calling program for each routine.

Program Name	Purpose
Main program	Reads parameters
CANRDG	Primary control program--reads Σ, k-values, contains sampling loop and k-value loops. Called by main program.
Subroutines called by CANRDG	
COR	Compute sample correlation matrix
EST	Compute population or sample canonical analysis given T or $\hat{T}$
FLIP	Transpose a matrix
GGNRM,GGNRM1	IMSL routine to generate multivariate normal samples
ITIMER	Cornell-HASP routine for timing (not required).
MSEBR	Gather and print statistics.
PFI	Partition correlation matrix and obtain triangular factors of diagonal blocks.
RLGQMI	IMSL routine to compute mean squares and products.
SSM	Move a symmetric matrix into symmetric storage mode, i.e., into a vector with the elements in the order (1,1), (1,2), (2,2), (1,3), (2,3), (3,3), ...

TRINV — Obtain the inverse of a triangular matrix in symmetric storage mode.

UPDATE — Compute Cholesky decomposition of A + D*I.

WRAY — Print one-dimensional array.

WRT2 — Print two-dimensional array.

Subroutines called by EST

BSIG — Computes H = F'G where G is in symmetric storage mode.

BTR — Computes $C = A'BD^{-1}$ where D is a diagonal matrix stored as a vector.

EHOBKS — IMSL routine to obtain eigen vectors of a symmetric matrix from those of a tridiagonal matrix obtained from the symmetric matrix by Householder transformations ( ).

EHOUSS — IMSL routine to tridiagonalize a symmetric matrix using Householder transformation.

EQRT2S — IMSL routine to obtain eigen values and eigen vectors of a tridiagonal matrix by LR transformations ( ).

PRODAA — Computes $T = (\Sigma_{11}^{-\frac{1}{2}} \Sigma_{12} \Sigma_{22}^{-\frac{1}{2}})(\Sigma_{11}^{-\frac{1}{2}} \Sigma_{12} \Sigma_{22}^{-\frac{1}{2}})'$ or $\hat{T}(k_1,k_2)$.

Subroutines called by MSEBR

M3 — Computes D = ABC' where A and C are imbedded in a 4-dimensional array as the last two dimensions. Used in computation of the true correlations of the average sample canonical variates.

Subroutines called by PFI

LUDCEP — IMSL routine for Cholesky decomposition of a symmetric matrix in symmetric storage mode.

PART — Partitions a matrix B into B_{11}, B_{12}, B_{22} where B_{11} and B_{22} are in symmetric storage mode.

Program Input

The program input consists of the following cards in the order shown:

I. Parameter Card. Read by main program.

Columns	Format Code	Information
1-5	I5	N, sample size for each multivariate sample.
6-10	I5	P, total number of variates.
11-15	I5	P1, number of variates in first set.
16-20	I5	P2, number of variates in the second set.
21-25	I5	M, number of multivariate N X P samples generated.
26	I1	SW1. If not 0 or blank mean square errors will be based upon the eigen vectors of $\hat{T}(k_1,k_2)$ rather than the canonical variates.
27	I1	SW2. If SW2 = 2 sample correlation matrices will be printed for each sample.
28	I1	SW3. If not 0 or blank the program will expect to run again with new data after the current run is complete.
29-31	I3	IPER. If 0 < IPER < M cumulative averages and mean square errors will be printed after every IPER-th sample, and after M samples; otherwise only after M samples.

II. Population correlation matrix. Read by CANRDG. Data is read row-wise with format code 10F8,6. There may be up to 10 entries per card but each row must begin a new card. If the decimal point is not punched 6 decimal places will be assumed.

III. Values for incrementing the diagonal of $\hat{\Sigma}_{11}$ and $\hat{\Sigma}_{22}$, read by CANRDG. The first card of this group contains K1, the number of increments for $\hat{\Sigma}_{11}$, in columns 1-5, format code I5. Subsequent cards contain the increments in columns 1-10, 11-20, ..., 71-80 with format code 10F10.5, as many cards as needed to supply K1 values. The next card contains K2, the number of increments for $\hat{\Sigma}_{22}$ in columns 1-5 with format code I5, and subsequent cards, the K2 increments, format 10F10.5. Note that the increment values of zero should not be supplied since they are automatically produced by the program (i.e., the program gives the usual $\hat{T}(0,0)$ canonical analysis as well as the ridge estimates).

Program Output

I. Printout of problem parameter values input: N, P, P1, P2, M, SEED.

II. Population canonical analysis.

Label	Content
CANONICAL CORRELATIONS, POPULATION	The population canonical correlations. These are really the square roots of the eigen values.
EIGENVECTORS	The eigen vectors of T, i.e., $B_1^{*'}$. The columns of the printed matrix are the eigen vectors. The first corresponds to the smallest root, etc.
B1P	The population canonical variates, i.e., B_1. The first row is the last canonical variate, etc.
B2*T	$B_2^{*'}$. The columns are the orthogonal vectors constituting the first p_1 rows of B_2^{*}.
B2P	The population variates for the second set of variates (in increasing order of the canonical correlations).

Squared lengths of B1 vectors. Lengths of the population canonical variates for the first set.

Squared lengths of B2 vectors. Lengths of the population canonical variates for the second set.

III. Sample output.

For each k_1,k_2 pair the average coefficients for each canonical variate, the mean square error and an adjusted mean square error [1] are printed. These are labeled B1 for the first set, B2 for the second set. As always the last canonical variate is first, the first canonical variate last. Following these two arrays the canonical correlation data is printed. This consists of four rows of output containing, respectively, the average canonical correlations, the mean squares, the minimum observed and the maximums observed. All the above is repeated for each k_1,k_2 pair, cumulatively after every IPER-th sample (see input section above) and after the M-th sample.

Finally after the average and mean square error output, the true correlations of the average sample canonical variates are printed.

References

[1] Anderson, D. A. and E. J. Carney (1974). Ridge regression estimation procedures applied to canonical correlation analysis. BU-509-M in the Mimeo Series, Biometrics Unit, Cornell University, Ithaca, N.Y.

[2] Hoerl, A. E. and R. W. Kennard (1970). Ridge regression: biased estimation for nonorthogonal problems. Technometrics 12, 55-66.

[3] Hoerl, A. E. and R. W. Kennard (1970). Ridge regression: applications to nonorthogonal problems. Technometrics 12, 69-82.

[4] International Mathematical Statistical Libraries, Inc. (1973). Computer Subroutine Libraries in Mathematics and Statistics (Abilities). Houston, Texas.

[5] Thorndike, R. M. and D. J. Weiss (1973). A study of the stability of canonical components. Educational and Psychological Measurement, 33, 123-184.

[6] Vinod, H. D. (1973). Canonical ridge and economics of joint production. Bell Laboratories Technical Memorandum, October 1973. Bell Laboratories, Holmdel, New Jersey.

```
****FORT     SYSPRINT ************RM3.PR1 ***FMT <00>* ****1234 LINES ***  2:07.54PM

TRAN IV G LEVEL  21                  MAIN              DATE = 74178          13/44/16

01                DOUBLE PRECISION SEED
02                INTEGER P1,P2,P
03                INTEGER SW1,SW2,SW3
04          1     READ(5,100)N,P,M,P1,P2,SEED,SW1,SW2,SW3,IPER
05          100   FORMAT(5I5,F15.13,3I1,I3)
            C   INPUT PARAMETERS
            C
            C   N     NUMBER OF OBSERVATIONS FOR EACH MULTIVARIATE SAMPLE
            C   P     NUMBER OF VARIABLES
            C   M     NUMBER OF MULTIVARIATE SAMPLES
            C   IF M=0 ONLY THE POPULATION CANONICAL ANALYSIS IS PERFORMED
            C   P1    NUMBER OF VARIABLES IN THE FIRST GROUP
            C   P2    NUMBER OF VARIABLES IN THE SECOND GROUP
            C   P1 LESS THAN OR EQUAL TO P2, P1 + P2 = P
            C   SEED DOUBLE PRECISION STARTING VALUE FOR THE RANDOM NUMBER
            C         GENERATOR.  0.0 < SEED < 1.0.
            C   SW1 NOT  ZERO  --MSE BASED ON B1*, B2*
            C   SW2=2    SAMPLE CORRELATIONS PRINTED
            C   SW3 NOT  ZERO  --RUN AGAIN WITH NEW DATA
            C   IPER < 1:  SUMMARY STATISTICS AT END OF RUN
            C   U <= IPER < M:  CUMULATIVE SUMMARY STATISTICS EVERY IPER-TH SAMPLE
06                WRITE(6,150) N,P,M,P1,P2,SEED
07          150   FORMAT(T5,'NUMBER IN SAMPLE',I5,/T5,'NUMBER OF VARIABLES',I5,
                  1/T5,'NUMBER OF SAMPLES',I5,/T5,'NUMBER X VARIABLES',I5,
                  2/T5,'NUMBER Y VARIABLES',I5,/T5,'SEED',F20.13)
08                MP=P*(P+1)/2
09                MPP=MP+P
10                MP1=P1*(P1+1)/2
11                MP2=P2*(P2+1)/2
12                      CALL CANRDG(N,P,M,P1,P2,MP,MP1,MP2,SEED,SW1,SW2,SW3,IPER)
13                WRITE(6,200)SEED
14          200   FORMAT('0',T50,'FINAL SEED= ',F19.16)
15                IF(SW3.NE.0)GO TO 1
16                END
```

```
01                SUBROUTINE CANRDG(N,P,M,P1,P2,MP,MP1,MP2,SEED,SW1,SW2,SW3,IPER)
          C    MAIN CONTROL PROGRAM
02                DOUBLE PRECISION SEED
03                INTEGER P1,P2,P
04                INTEGER SW1,SW2,SW3
05                INTEGER OUT
          C    DIMENSIONS OF THE FOLLOWING ARRAYS MUST FIT THE VALUES OF INPUT
          C    PARAMETERS AS SHOWN BELOW:
          C    SA(P,P), SIGMA(P(P+1)/2), B1STAR(P1,P1), B2STAR(P2,P2), B1(P1,P1),
          C    B2(P2,P2), DP(P1), XY(N,P+P(P+1)/2), XBAR(P+P(P+1)/2), DS(P1),
          C    R11(P1,P1), R12(P1,P2), R22(P2,P2), RT11(P1(P1+1)/2), RT22(P2(P2+1)/2
          C    R(P1(P1+1)/2), RI(P1(P1+1)/2), S(P2(P2+1)/2), SI(P2(P2+1)/2), T(P),
          C    RZ(P1(P1+1)/2), SZ(P2(P2+1)/2), A(P1,P2), AA(P1(P1+1)/2),
          C    RZ(P1(P1+1)/2), SZ(P2(P2+1)/2), A(P1,P2), AA(P1(P1+1)/2),
06                DIMENSION SA(10,10),SIGMA(55)
07                DIMENSION B1STAR(5,5),B2STAR(5,5),B1(5,5),B2(5,5),DP(5)
08                DIMENSION XBAR(65),DS(5)
09                DIMENSION R11(5,5),R12(5,5),R22(5,5),RT11(15),RT22(15),
                 1R(15),RI(15),S(15),SI(15),T(10),RZ(15),SZ(15),A(5,5),AA(15)
10                DIMENSION DK1(50),DK2(50),DQ(5),DT(5)
11                DIMENSION B1P(5,5),B2P(5,5)
12                DIMENSION RS(5,5)
13                DIMENSION XY(50,65)
14                OUT=6
15                K1=0
16                K2=0
17                K11=1
18                K21=1
19                DN=1./N
20                IF(SW2.EQ.3)GO TO 1
21                IF(IPER.LE.0)IPER=M
22                KKT=IPER+1
          C    READ COVARIANCE MATRIX
23                READ(5,105)((SA(I,J),I=1,P),J=1,P)
24          105   FORMAT(10F8.6)
          C    SIGMA = SA SYMMETRIC STORAGE MODE
25                CALL SSM(SA,SIGMA,P)
26                CALL PFI(SA,RT11,R12,RT22,R,S,P,P1,P2,MP1,MP2)
27                DO 600 I=1,P1
28                DO 600 J=1,P1
29                R11(I,J)=SA(I,J)
30          600   CONTINUE
31                DO 601 I=1,P2
32                II=I+P1
33                DO 601 J=1,P2
34                JJ=J+P1
35                R22(I,J)=SA(II,JJ)
36          601   CONTINUE
37                DO 602 I=1,P1
38                DO 602 J=1,P2
39                RS(I,J)=R12(I,J)
40          602   CONTINUE
          C    INVERT R, S
41                CALL TRINV(R,RI,P1)
42                CALL TRINV(S,SI,P2)
          C    CANONICAL ANALYSIS
43                CALL EST(R,RI,S,SI,R12,A,AA,B1STAR,B2STAR,B1P,B2P,DP,DQ,T,
                 *P,P1,P2,MP1,MP2)
```

```
      C   DP ARE THE SQUARED CANONICAL CORRELATIONS FOR THE POPULATION
      C   B1P ARE THE POPULATION COEFFICIENTS OF THE CANONICAL VARIATES FOR
      C   FIRST GROUP; B2P, FOR THE SECOND GROUP.  THESE ARE IN INCREASING
      C   ORDER OF THE CANONICAL CORRELATIONS, SO THAT THE FIRST CANONICAL
      C   VARIATE IS LAST, THE SECOND, NEXT TO LAST, ETC.
44              WRITE(6,902)DP
45              WRITE(6,903)B1STAR
46        902   FORMAT(///,T15,'CANONICAL CORRELATIONS',/(5X,5F20.8))
47        903   FORMAT(///,T15,'EIGENVECTORS',/(5X,5F20.8))
48              CALL WRT2('B1P ',P1,P1,B1P)
49              CALL WRT2('B2*T',P2,P1,B2STAR)
50              CALL WRT2('B2P ',P2,P1,B2P)
      C   $$$$$
51              IF (M.LE.0) RETURN
52              IF (SW1.EQ.0 ) GO TO 1000
53              CALL FLIP(B1STAR,B1P,P1,P1)
54              CALL FLIP(B2STAR,B2P,P1,P2)
55        1000  CONTINUE
      C   M P-VARIATE SAMPLES OF SIZE N
      C   READ K VALUES
      C   K1 IS THE NUMBER OF K-VALUES WHICH WILL BE USED TO INCREMENT THE
      C   DIAGONAL ELEMENTS OF THE FIRST GROUP MATRIX, K2 THE NUMBER OF VALUES
      C   FOR THE SECOND GROUP.  THESE VALUES ARE READ INTO DK1 AND DK2
      C   RESPECTIVELY.  IF K1 OR K2 IS GREATER THAN 50 THE DIMENSION OF
      C   DK1 AND DK2 MUST BE INCREASED.
56        129   FORMAT(I5)
57        130   FORMAT(8F10.5)
58              READ(5,129)K1
59              READ(5,130)(DK1(I),I=1,K1)
60              READ(5,129)K2
61              READ(5,130)(DK2(I),I=1,K2)
62              K11=K1+1
63              K21=K2+1
64              IF(M.EQ.1) GO TO 1
      C   INITIALIZE MSEBR
      C   INIT =0  INITIALIZE
      C   INIT=1  UPDATE ARRAYS FOR CURRENT SAMPLE
      C   INIT=2  PRINT SUMMARY
      C   INIT=3  COMPUTE AND PRINT CANNO
      C   INIT=3  COMPUTE AND PRINT CANONICAL CORRELATIONS OF AVERAGE
      C   SAMPLE CANONICAL VARIATES WITH RESPECT TO  THE POPULATION
      C   VARIANCE-COVARIANCE MATRIX.
65              INIT=0
66                    CALL MSEBR(B1,B2,B1P,B2P,DS,DP,DK1,DK2,K11,K21,P1,P2,
           *INIT,OUT,R11,R22,RS)
67              INIT=1
      C   INITIALIZE GGNRM
68              CALL GGNRM(SEED,1,P,SIGMA,N,XY,T,IER)
      C   BEGIN OUTSIDE LOOP FOR M  NXP SAMPLES
69        1      CONTINUE
      C   ITIMER IS A CORNELL HASP TIMER.  ITS INCLUSION HERE HAS NO EFFECT
      C   ON THE COMPUTATIONS BUT CAUSES THE TIME PER SAMPLE TO BE PRINTED.
70              CALL ITIMER(0)
71              DO 500 KOUT=1,M
72              IF(KOUT.NE.KKT)GO TO 14
73              KKT1=KKT-1
74              WRITE(OUT,402)KKT1
75        402   FORMAT('0',T5,'AFTER SAMPLE NO. ',I3)
```

```
76                KKT=KKT+IPER
77                INIT=2
78                      CALL MSEBR(B1,B2,B1P,B2P,DS,DP,DK1,DK2,K11,K21,P1,P2,
                 *INIT,OUT,R11,R22,RS)
79                INIT=1
80                CALL ITIMER(1)
81          14    CONTINUE
82                IF(SW2.NE.3)GO TO 15
83                READ(5,401)((XY(I,J),J=1,P),I=1,N)
84                GO TO 16
85          15    CONTINUE
86                IF(M.EQ.1) GO TO 4
87                CALL GGNRM1(SEED,N,P,SIGMA,N,XY,T,IER)
88          16    CONTINUE
            C     COMPUTE XBAR
89                DO 3 J=1,P
90                U=0.0
91                DO 2 I=1,N
92                U=U+XY(I,J)
93          2     CONTINUE
94                XBAR(J)=U*DN
95          3     CONTINUE
            C     FORM CORRELATION MATRIX
96                CALL RLGQMI (XY,N,P,N,XBAR)
97                CALL COR (P,XBAR,SA)
            C     SAMPLE CORRELATIONS
98                IF(SW2.NE.2.AND.SW2.NE.3)GO TO 4
99                CALL WRT2('CORR',P,P,SA)
00          4      CONTINUE
01          401   FORMAT(5F10.7)
            C     PARTITION AND FACTOR
02                CALL PFI(SA,RT11,R12,RT22,R,S,P,P1,P2,MP1,MP2)
            C     STORE INITIAL FACTORS
03                DO 20 I=1,MP2
04          20    SZ(I)=S(I)
            C
            C     BEGIN K1 LOOP
05                DO 80 I1=1,K11
06                D1=0.0
07                IF(I1.EQ.1)GO TO 60
08                D1=DK1(I1-1)
09                CALL UPDATE(RT11,R,D1,P1,MP1)
10          60    CALL TRINV(R,RI,P1)
            C     BEGIN K2 LOOP
11                DO 70 I2=1,K21
12                IF(I2.GT.1) GO TO 65
13                D2=0.0
14                DO 62 II=1,MP2
15          62    S(II)=SZ(II)
16                GO TO 68
17          65    D2=DK2(I2-1)
18                CALL UPDATE(RT22,S,D2,P2,MP2)
19          68    CALL TRINV(S,SI,P2)
            C     SAMPLE CANONICAL ANALYSIS
20                CALL EST(R,RI,S,SI,R12,A,AA,B1STAR,B2STAR,B1,B2,DS,DT,T,P,P1,P2,
                 *MP1,MP2)
            C     $$$$$
            C     $$$$$
```

```
IRAN IV G LEVEL  21                    CANRDG              DATE = 74178          13/44/16

21                IF(M.EQ.1) GO TO 70
22                IF ( SW1.EQ.0 ) GO TO 2000
23                CALL FLIP(B1STAR,B1,P1,P1)
24                CALL FLIP(B2STAR,B2,P1,P2)
25          2000  CONTINUE
            C    COLLECT STATISTICS
26                      CALL MSEBR(B1,B2,B1P,B2P,DS,DP,DK1,DK2,I1, I2, P1,P2,
                 *INIT,OUT,R11,R22,RS)
27          70    CONTINUE
28          80    CONTINUE
29          500   CONTINUE
30                IF(M.EQ.1)RETURN
31                IF(SW1.NE.0) WRITE(6,904)
32          904   FORMAT('0******',T15,'MEAN SQUARE ERROR BASED ON EIGEN VECTORS')
33                INIT=3
34                      CALL MSEBR(B1,B2,B1P,B2P,DS,DP,DK1,DK2,K11,K21,P1,P2,
                 *INIT,OUT,R11,R22,RS)
35                RETURN
36                END
```

```
TRAN IV G LEVEL  21                    COR                 DATE = 74178          13/44/16

01                SUBROUTINE COR(P,X,R)
            C    COMPUTES CORRELATIONS FROM AVERAGE CROSS PRODUCTS GIVFN BY IMSL
            C    ROUTIN RLGQMI.
02                INTEGER P
03                REAL*4 X(1),R(P,P)
04                DO 1 I=1,P
05                K=I+P
06                X(K)=1./SQRT(X(K))
07          1     CONTINUE
08                K=2*P
09                DO 3 I=1,P
10                R(I,I)=1.
11                KI=P+I
12                D=X(KI)
13                DO 2 J=I,P
14                IF(I.EQ.J)GO TO 2
15                KJ=P+J
16                K=K+1
17                R(I,J)=X(K)*X(KJ)*D
18                R(J,I)=R(I,J)
19          2     CONTINUE
20          3     CONTINUE
21                RETURN
22                END
```

```
TRAN IV G LEVEL  21                    EST                    DATE = 74178          13/44/16

01                SUBROUTINE EST(R,RI,S,SI,R12,A,AA,B1STAR,B2STAR,B1,B2,DS,DT,T,P,
                 *P1,P2,MP1,MP2)
02                INTEGER P,P1,P2
03                REAL*4 R(MP1),RI(MP1),S(MP2),SI(MP2),R12(P1,P2),A(P1,P2),AA(MP1),
                 *B1STAR(P1,P2),B2STAR(P2,P2),B1(P1,P1),B2(P2,P2),DS(P1),DT(P1),T(P)
          C     FORM PRODUCT (RI*R12*SI)*(          )'
04                CALL PRODAA(RI,R12,SI,A,AA,P1,P2,MP1,MP2)
          C     COMPUTE EIGENVALUES AND EIGENVECTORS OF AA'
          C     REDUCE TO TRIDIAGONAL
05                CALL EHOUSS(AA,P1,DS,DT,T)
          C     COMPUTE EIGENVALUES AND EIGENVECTORS OF TRIDIAGONAL MATRIX
06                DO 10 I=1,P1
07                DO 10 J=1,P1
08                B1STAR(I,J)=0.0
09                IF(I.EQ.J)B1STAR(I,J)=1.0
10          10    CONTINUE
11                CALL EQRT2S(DS,DT,P1,B1STAR,P1,IER)
          C     BACK TRANSFORMATION
12                CALL EHOBKS(AA,P1,1,P1,B1STAR,P1)
13                CALL BSIG(B1STAR,RI,B1,P1,P1,MP1)
14                DO 12 I=1,P1
15                IF(DS(I).LT.0.0)DS(I)=0.0
16                DS(I)=SQRT(DS(I))
17          12    CONTINUE
18                CALL BTR(A,B1STAR,B2STAR,DS,P1,P2)
19                CALL BSIG(B2STAR,SI,B2,P1,P2,MP2)
20                RETURN
21                END
```

```
TRAN IV G LEVEL  21                    FLIP                   DATE = 74178          13/44/16

01                SUBROUTINE FLIP(B,C,P,Q)
          C     COMPUTE C=B-TRANSPOSE.
02                INTEGER P,Q
03                DIMENSION B(P,Q),C(Q,P)
04                DO 2 I=1,P
05                DO 1 J=1,Q
06                C(J,I)=B(I,J)
07          1     CONTINUE
08          2     CONTINUE
09                RETURN
10                END
```

```
01                SUBROUTINE MSEBR(B1,B2,B1P,B2P,DS,DP,DK1,DK2,I1, I2, P1,P2,
                 *INIT,OUT,R11,R22,RS)
          C     COMPUTES MEAN SQUARE ERRORS AND AVERAGES OF THE CANONICAL VARIATES
          C     AND CANONICAL CORRELATIONS.
          C     ARRAYS  ARE DIMENSIONED AS FOLLOWS:  (KX=K1+1, KY=K2+1)
          C     QR(KX,KY,P1), QB(KX,KY,P1), QC(KX,KY,P1), AR(KX,KY,P1), AB(KX,KY,P1,P1
          C     DK1(K1), DK2(K2), RR1(KX,KY,P1,P1), RR2(KX,KY,P1,P1)
          C     RMN(KX,KY,P1), RMX((KX,KY,P1)
          C     TT( MAX(P1,P2), MAX(P1,P2) ), D1(P1), D2(P2).
02                INTEGER P1,P2,OUT
03                DIMENSION B1(P1,P1),B2(P1,P2),B1P(P1,P1),B2P(P1,P2),DS(P1),DP(P1)
04                DIMENSION R11(P1,P1), RS(P1,P2), R22(P2,P2)
05                DIMENSION QR(10,10,5),QB(10,10,5),QC(10,10,5),AR(10,10,5),
                 *AB(10,10,5,5),AC(10,10,5,5)
06                DIMENSION DK1(50),DK2(50)
07                DIMENSION RR1(10,10,5),RR2(10,10,5),RMN(10,10,5),RMX(10,10,5),
                 *SP1(5),SP2(5)
08                DIMENSION TT(5,5),D1(5),D2(5)
09                IF (INIT.NE.0)GO TO 100
10                MD=I1*I2
11                Z=1.0
12                W=0.0
13                DO 4 I=1,I1
14                DO 4 J=1,I2
15                DO 3 K=1,P1
16                AR(I,J,K)=0.0
17                QR(I,J,K)=0.0
18                QB(I,J,K)=0.0
19                QC(I,J,K)=0.0
20                RR1(I,J,K)=0.0
21                RR2(I,J,K)=0.0
22                RMN(I,J,K)=1.0
23                RMX(I,J,K)=0.0
24                DO 2 L=1,P1
25                AB(I,J,K,L)=0.0
26          2     CONTINUE
27                DO 3 L=1,P2
28                AC(I,J,K,L)=0.0
29          3     CONTINUE
30          4     CONTINUE
31                DO 55 I=1,P1
32                T=0.0
33                DO 5 J=1,P1
34                T=T+B1P(I,J)**2
35          5     CONTINUE
36                SP1(I)=T
37                T=0.0
38                DO 51 J=1,P2
39                T=T+B2P(I,J)**2
40          51    CONTINUE
41                SP2(I)=T
42          55    CONTINUE
43                WRITE(OUT,210)(SP1(I),I=1,P1)
44                WRITE(OUT,211)(SP2(I),I=1,P1)
45          210   FORMAT('0',T5,'SQUARED LENGTH OF B1 VECTORS',/(5F20.8))
46          211   FORMAT('0',T5,'SQUARED LENGTH OF B2 VECTORS',/(5F20.8))
47          1     CONTINUE
48                RETURN
```

```
      100   CONTINUE
            IF (INIT.NE.1)GO TO 200
            DO 16 I=1,P1
            IF(DS(I).LT.RMN(I1,I2,I))RMN(I1,I2,I)=DS(I)
            IF(DS(I).GT.RMX(I1,I2,I))RMX(I1,I2,I)=DS(I)
            RR1(I1,I2,I)=W*RR1(I1,I2,I)
            RR2(I1,I2,I)=W*RR2(I1,I2,I)
            AR(I1,I2,I)=W*AR(I1,I2,I)
            AR(I1,I2,I)=AR(I1,I2,I)+DS(I)/Z
            QR(I1,I2,I)=QR(I1,I2,I)*W
            QR(I1,I2,I)=QR(I1,I2,I)+((DS(I)-DP(I))**2)/Z
            QB(I1,I2,I)=W*QB(I1,I2,I)
            QC(I1,I2,I)=W*QC(I1,I2,I)
            T=0.0
            U=0.0
            V=0.0
            S=0.0
            DO 6 J=1,P1
            AB(I1,I2,I,J)=W*AB(I1,I2,I,J)
            T=T+(B1(I,J)-B1P(I,J))**2
            U=U+(B1(I,J)+B1P(I,J))**2
            V=V+B1(I,J)*B1P(I,J)
            S=S+B1(I,J)**2
      6     CONTINUE
            IF(T.LT.U) GO TO 8
            T=U
            DO 7 J=1,P1
            AB(I1,I2,I,J)=AB(I1,I2,I,J)-B1(I,J)/Z
      7     CONTINUE
            GO TO 10
      8     DO 9 J=1,P1
            AB(I1,I2,I,J)=AB(I1,I2,I,J)+B1(I,J)/Z
      9     CONTINUE
      10    QB(I1,I2,I)=QB(I1,I2,I)+(T/SP1(I))/Z
            RR1(I1,I2,I)=RR1(I1,I2,I)+(1.-V**2/(S*SP1(I)))/Z
            T=0.0
            U=0.0
            V=0.0
            S=0.0
            DO 11 J=1,P2
            T=T+(B2(I,J)-B2P(I,J))**2
            U=U+(B2(I,J)+B2P(I,J))**2
            AC(I1,I2,I,J)=W*AC(I1,I2,I,J)
            V=V+B2(I,J)*B2P(I,J)
            S=S+B2(I,J)**2
      11    CONTINUE
            IF(T.LT.U)GO TO 13
            T=U
            DO 12 J=1,P2
            AC(I1,I2,I,J)=AC(I1,I2,I,J)-B2(I,J)/Z
      12    CONTINUE
            GO TO 15
      13    DO 14 J=1,P2
            AC(I1,I2,I,J)=AC(I1,I2,I,J)+B2(I,J)/Z
      14    CONTINUE
      15    QC(I1,I2,I)=QC(I1,I2,I)+(T/SP2(I))/Z
            RR2(I1,I2,I)=RR2(I1,I2,I)+(1.-V**2/(S*SP2(I)))/Z
      16    CONTINUE
```

```
07                IF(I1*I2.LT.MD)GO TO 1
08                Z=Z+1.0
09                W=(Z-1.0)/Z
10                GO TO 1
11          200   CONTINUE
12          201   FORMAT('1',T15,'AVERAGES AND MEAN SQUARE ERRORS')
13                WRITE(OUT,201)
14                T=0.0
15                U=0.0
16                DO 22 I=1,I1
17                IF(I.EQ.1)GO TO 17
18                T=DK1(I-1)
19                U=0.0
20          17    DO 21 J=1,I2
21                IF(J.EQ.1)GO TO 18
22                U=DK2(J-1)
23          18    WRITE(OUT,202)T,U
24          202   FORMAT(///,T15,'K1=',F6.3,T25,'K2=',F6.3)
25                WRITE(OUT,203)
26                DO 19 K=1,P1
27                WRITE(OUT,204)(AB(I,J,K,L),L=1,P1),QB(I,J,K),RR1(I,J,K)
28          203   FORMAT('0',T15,'B1')
29          204   FORMAT(T5,5F15.8,10X,2F15.8)
30          19    CONTINUE
31                WRITE(OUT,205)
32          205   FORMAT('0',T15,'B2')
33                DO 20 K=1,P1
34                WRITE(OUT,204)(AC(I,J,K,L),L=1,P1),QC(I,J,K),RR2(I,J,K)
35          20    CONTINUE
36                WRITE(OUT,206)
37          206   FORMAT('0',T15,'CORRELATIONS/MSE')
38                WRITE(OUT,207)(AR(I,J,K),K=1,P1),(QR(I,J,K),K=1,P1)
39          207   FORMAT(T5,5F20.8)
40                WRITE(OUT,208)(RMN(I,J,K),K=1,P1)
41          208   FORMAT(T2,'MIN',5F20.8)
42                WRITE(OUT,209)(RMX(I,J,K),K=1,P1)
43          209   FORMAT(T2,'MAX',5F20.8)
44          21    CONTINUE
45          22    CONTINUE
46                IF(INIT.LE.2)GO TO 1
47                T=0.0
48                DO 350 I=1,I1
49                IF(I.EQ.1)GO TO 300
50                T=DK1(I-1)
51          300   U=0.0
52                DO 340 J=1,I2
53                IF(J.EQ.1)GO TO 301
54                U=DK2(J-1)
55          301   CALL M3(AB,R11,AB,TT,10,10,P1,P1,I,J)
56                DO 302 K=1,P1
57                D1(K)=1./SQRT(TT(K,K))
58          302   CONTINUE
59                CALL M3(AC,R22,AC,TT,10,10,P2,P2,I,J)
60                DO 303 K=1,P2
61                D2(K)=1./SQRT(TT(K,K))
62          303   CONTINUE
63                CALL M3(AB,RS,AC,TT,10,10,P1,P2,I,J)
64                DO 310 K=1,P1
```

```
TRAN IV G LEVEL  21                    MSEBR                    DATE = 74178          13/44/16

65                DO 310 L=1,P1
66                TT(K,L)=TT(K,L)*D1(K)*D2(L)
67          310   CONTINUE
68                WRITE(OUT,399)T,U,((TT(K,L),L=1,P1),K=1,P1)
69          399   FORMAT('0',T5,'K1=',F6.3,5X,'K2=',F6.3,
                 */(T5,5F15.8))
70          340   CONTINUE
71          350   CONTINUE
72                GO TO 1
73                END
```

```
TRAN IV G LEVEL  21                    PFI                      DATE = 74178          13/44/16

01                SUBROUTINE PFI(SA,RT11,R12,RT22,R,S,P,P1,P2,MP1,MP2)
02                INTEGER P,P1,P2
03                REAL*4 SA(P,P),RT11(MP1),R12(P1,P2),RT22(MP2),R(MP1),S(MP2)
          C     PARTITION SA
04                CALL PART(SA,P,RT11,MP1,R12,P1,P2,RT22,MP2)
          C     FACTOR RT11
05                CALL LUDECP(RT11,R,P1,DX,DN,IER)
          C     FIX DIAGONAL
06                II=0
07                DO 30 I=1,P1
08                II=II+I
09                R(II)=1.0/R(II)
10          30    CONTINUE
          C     FACTOR RT22
11                CALL LUDECP(RT22,S,P2,DX,DN,IER)
          C     FIX DIAGONAL
12                II=0
13                DO 40 I=1,P2
14                II=II+I
15                S(II)=1.0/S(II)
16          40    CONTINUE
17                RETURN
18                END
```

```
TRAN IV G LEVEL  21                    SSM                      DATE = 74178          13/44/16

01                SUBROUTINE SSM(A,B,N)
          C     CONVERT A(N,N) INTO SYMMETRIC STORAGE MODE IN B(N(N+1)/2).
02                DIMENSION A(N,N),B(1)
03                K=0
04                DO 1 I=1,N
05                DO 1 J=1,I
06                K=K+1
07                B(K)=A(I,J)
08          1     CONTINUE
09                RETURN
10                END
```

```
TRAN IV G LEVEL  21                    TRINV                DATE = 74178          13/44/16

01                SUBROUTINE TRINV(R,S,N)
          C    INVERT THE NXN TRIANGULAR MATRIX R TO GET S
          C    STORAGE IN COLUMN ORDER (1,1) (1,2) (2,2) (1,3) (2,3)...
02                REAL*4 R(1),S(1),T
03                INTEGER N
04                I=N+1
05                II=I*(N+2)/2
06                DO 3 IN=1,N
07                I1=I
08                I=I-1
09                II=I1-I1
10                S(II)=1.0/R(II)
11                IJ=II
12                DO 2 J=I,N
13                IF(I.EQ.J) GO TO 2
14                IJ=IJ+J-1
15                KJ=IJ
16                IK=II
17                T=0.0
18                DO 1 K=I1,J
19                IK=IK+K-1
20                KJ=KJ+1
21                T=T-R(IK)*S(KJ)
22          1     CONTINUE
23                S(IJ)=T*S(II)
24          2     CONTINUE
25          3     CONTINUE
26                RETURN
27                END


TRAN IV G LEVEL  21                    UPDATE               DATE = 74178          13/44/16

01                SUBROUTINE UPDATE(A,U,D,N,N2)
          C    CHOLESKY DECOMPOSITION OF A + D*I
02                DIMENSION A(N2), U(N2)
          C    N2=N(N+1)/2
          C    INCREMENT DIAGONAL
03                II=0
04                DO 1 I=1,N
05                II=II+I
06                A(II)=A(II)+D
07          1     CONTINUE
          C     IMSL CHOLESKY ROUTINE
08                CALL LUDECP(A,U,N,DX,DN,IER)
          C    FIX DIAGONAL
09                II=0
10                DO 2 I=1,N
11                II=II+I
12                U(II)=1.0/U(II)
13          2     CONTINUE
14                RETURN
15                END
```

```
TRAN IV G LEVEL  21                    WRAY                    DATE = 74178          13/44/16

01                SUBROUTINE WRAY(NAME,ARRAY,SIZE)
02                INTEGER SIZE
03                DIMENSION ARRAY(SIZE)
04                WRITE(6,100)NAME,(ARRAY(I),I=1,SIZE)
05          100   FORMAT(///T15, A4,/(5X,5F20.8))
06                RETURN
07                END
```

```
TRAN IV G LEVEL  21                    WRT2                    DATE = 74178          13/44/16

01                SUBROUTINE WRT2(NAME,N,M,ARRAY)
02                REAL*4 ARRAY(N,M)
03                WRITE(6,100) NAME
04          100   FORMAT(///T15, A4)
05                DO 1 I=1,N
06                WRITE(6,101) I,(ARRAY(I,J),J=1,M)
07          101   FORMAT(T5,I5/(5X,5F20.8))
08          1     CONTINUE
09                RETURN
10                END
```

```
TRAN IV G LEVEL  21                    BSIG                    DATE = 74178          13/44/16

01                SUBROUTINE BSIG(F,G,H,P,Q,MQ)
          C   COMPUTES H=(F-TRANSPOSE)*G.  G IS IN SYMMETRIC STORAGE MODE
02                INTEGER P,Q
03                REAL*4 F(Q,P),G(MQ),H(P,Q)
04                DO 3 I=1,P
05                JJ=0
06                DO 2 J=1,Q
07                JJ=JJ+J
08                KJ=JJ
09                T=0.0
10                DO 1 K=J,Q
11                T=T+F(K,I)*G(KJ)
12                KJ=KJ+K
13          1     CONTINUE
14                H(I,J)=T
15          2     CONTINUE
16          3     CONTINUE
17                RETURN
18                END
```

```
01                SUBROUTINE BTR(A,B,C,D,P,Q)
          C    COMPUTES C=A'*B*D-INVERSE, D DIAGONAL
02                INTEGER P,Q
03                REAL*4 A(P,Q),B(P,P),C(Q,P),D(P)
04                DO 3 J=1,P
05                IF(D(J).GT.0) GO TO 10
06                WRITE(6,11)J
07          11    FORMAT(' ZERO CANONICAL CORRELATION.  J=',I2)
08                C(I,J)=0.0
09                GO TO 2
10          10    CONTINUE
11                T=1.0/D(J)
12                DO 2 I=1,Q
13                U=0.0
14                DO 1 K=1,P
15                U=U+A(K,I)*B(K,J)
16          1     CONTINUE
17                C(I,J)=T*U
18          2     CONTINUE
19          3     CONTINUE
20                RETURN
21                END
```

```
01                SUBROUTINE PRODAA(A,B,C,D,E,P1,P2,MP1,MP2)
          C     FORM PRODUCTS D=ABC', E=DD'
          C     B AND D ARE GENERAL P1XP2 MATRICES
          C     A, C, AND E ARE IN SYMMETRIC  STORAGE MODE, A AND C  TRIANGULAR,
          C     E SYMMETRIC.
          C     MPX=PX*(PX+1)/2
02                INTEGER P,P1,P2
03                DIMENSION A(MP1),B(P1,P2),C(MP2),D(P1,P2),E(MP1)
          C     COMPUTE D=ABC'
04                II=0
05                DO 4 I=1,P1
06                II=II+I
07                JJ=0
08                DO 3 J=1,P2
09                IK=II-I
10                JJ=JJ+J
11                U=0.0
12                DO 2 K=1,I
13                JL=JJ-J
14                IK=IK+1
15                S=A(IK)
16                T=0.0
17                DO 1 L=1,J
18                JL=JL+1
19                T=T+B(K,L)*C(JL)
20          1     CONTINUE
21                U=U+S*T
22          2     CONTINUE
23                D(I,J)=U
24          3     CONTINUE
25          4     CONTINUE
          C
          C     COMPUTE E=DD'=(ABC')(ABC')'
26                II=0
27                DO 6 I=1,P1
28                IJ=II
29                II=II+I
30                DO 6 J=1,I
31                IJ=IJ+1
32                V=0.0
33                DO 5 K=1,P2
34                V=V+D(I,K)*D(J,K)
35          5     CONTINUE
36                E(IJ)=V
37          6     CONTINUE
38                RETURN
39                END
```

```
TRAN IV G LEVEL  21                    M3                    DATE = 74178          13/44/16

01                SUBROUTINE M3(A,B,C,D,N1,N2,P1,P2,N,M)
          C   COMPUTES D=A*B*(C-TRANSPOSE)
02                INTEGER P1,P2
03                DIMENSION A(N1,N2,P1,P2),B(P2,P2),C(N1,N2,P1,P2),D(P1,P1)
04                DO 100 I=1,P1
05                DO 100 J=1,P1
06                T=0.0
07                DO 90 K=1,P2
08                U=0.0
09                DO 80 L=1,P2
10                U=U+B(K,L)*C(N,M,J,L)
11          80    CONTINUE
12                T=T+U*A(N,M,I,K)
13          90    CONTINUE
14                D(I,J)=T
15          100   CONTINUE
16                RETURN
17                END
```

```
TRAN IV G LEVEL  21                    PART                  DATE = 74178          13/44/16

01                SUBROUTINE PART(B,P,B11,MP1,B12,P1,P2,B22,MP2)
          C   PARTITIONS THE SYMMETRIC P X P MATRIX B INTO  B11, B12, B22.
          C   B11, B22 IN SYMMETRIC STORAGE MODE, P1(P1+1)/2, P2(P2+1)/2, RESPY.
02                INTEGER P,P1,P2
03                DIMENSION B(P,P),B11(MP1),B12(P1,P2),B22(MP2)
04                K=0
05                DO 1 I=1,P1
06                DO 1 J=1,I
07                K=K+1
08          1     B11(K)=B(I,J)
09                DO 2 I=1,P1
10                DO 2 J=1,P2
11                KJ=J+P1
12          2     B12(I,J)=B(I,KJ)
13                II=0
14                DO 4 I=1,P2
15                IJ=II
16                II=II+I
17                KI=I+P1
18                DO 3 J=1,I
19                IJ=IJ+1
20                KJ=P1+J
21          3     B22(IJ)=B(KI,KJ)
22          4     CONTINUE
23                RETURN
24                END
```